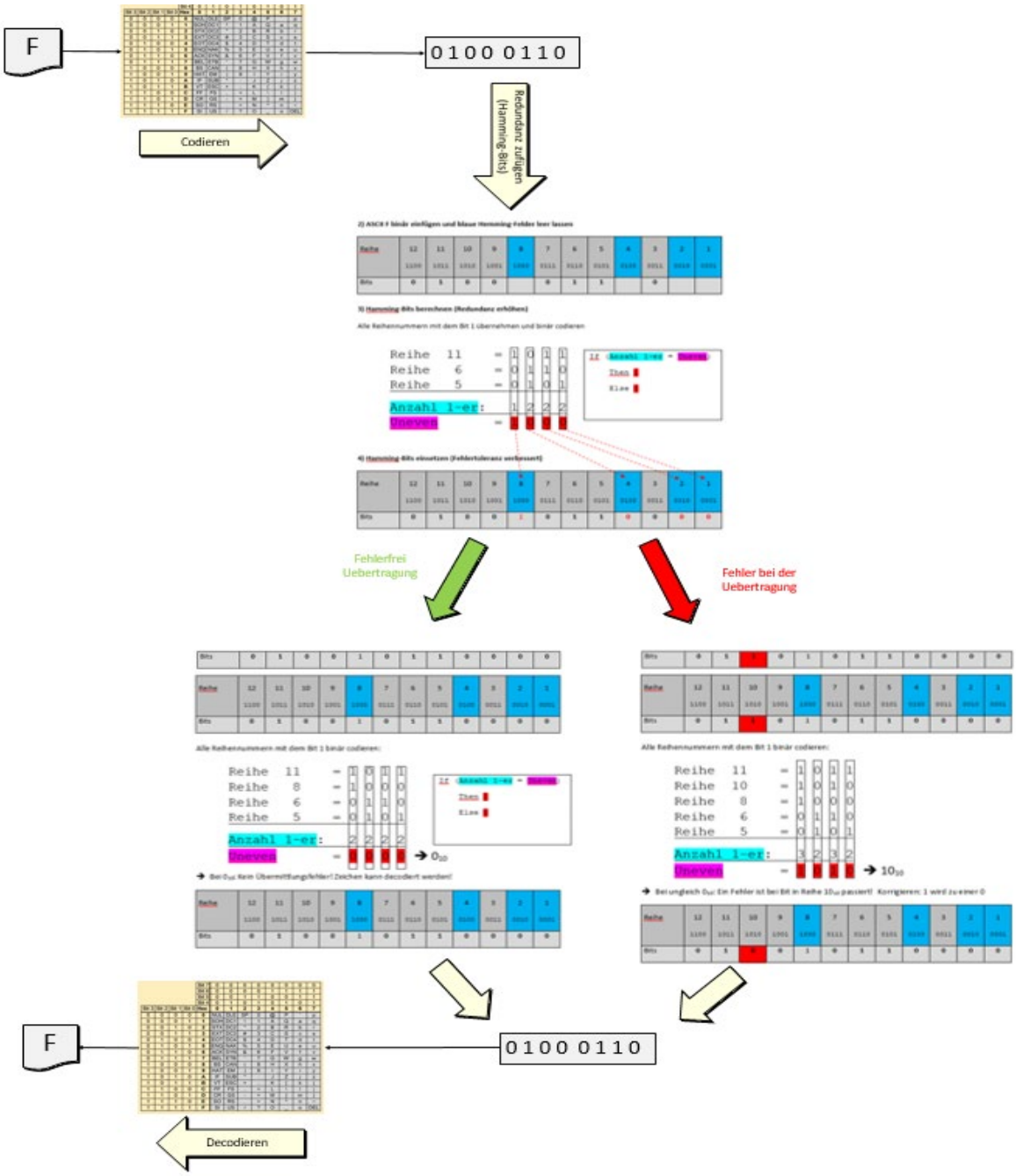


HAMMING-CODE

Ein Lösungsraster

Rothlin Walter
walter@rothlin.com

1) Nutzdaten codieren: F (z.B. ASCII-Tabelle)

Binär				Bit 7		0	0	0	0	0	0	0	0
				Bit 6		0	0	0	0	1	1	1	1
				Bit 5		0	0	1	1	0	0	1	1
				Bit 4		0	1	0	1	0	1	0	1
Bit 3	Bit 2	Bit 1	Bit 0	Hex		0	1	2	3	4	5	6	7
					Dezi	0	16	32	48	64	80	96	112
0	0	0	0	0	0	NUL	DLE		0	@	P	`	p
0	0	0	1	1	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	2	2	STX	DC2	"	2	B	R	b	r
0	0	1	1	3	3	EXT	DC3	#	3	C	S	c	s
0	1	0	0	4	4	EOT	DC4	$	4	D	T	d	t
0	1	0	1	5	5	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	6	6	ACK	SYN	&	6	F	V	f	v
0	1	1	1	7	7	BEL	ETB	'	7	G	W	g	w
1	0	0	0	8	8	BS	CAN	(	8	H	X	h	x
1	0	0	1	9	9	HAT	EM	)	9	I	Y	i	y
1	0	1	0	A	10	IF	SUB	*	:	J	Z	j	z
1	0	1	1	B	11	VT	ESC	+	;	K	[	k	{
1	1	0	0	C	12	FF	FS	,	<	L	\	l	|
1	1	0	1	D	13	CR	GS	-	=	M	]	m	}
1	1	1	0	E	14	SOH	RS	.	>	N	^	n	~
1	1	1	1	F	15	SI	US	/	?	O	_	o	DEL

ASCII	Binär	Hex	Dez
F	0100 ' 0110	46	70

Bin	Dez	Hex
0000	**00**	**0**
0001	**01**	**1**
0010	**02**	**2**
0011	**03**	**3**
0100	**04**	**4**
0101	**05**	**5**
0110	**06**	**6**
0111	**07**	**7**
1000	**08**	**8**
1001	**09**	**9**
1010	**10**	**A**
1011	**11**	**B**
1100	**12**	**C**
1101	**13**	**D**
1110	**14**	**E**

1111	15	F

ASCII F	0	1	0	0	0	1	1	0

2) ASCII F binär einfügen und blaue Hemming-Felder leer lassen

Reihe	12 1100	11 1011	10 1010	9 1001	8 1000	7 0111	6 0110	5 0101	4 0100	3 0011	2 0010	1 0001
Bits	**0**	**1**	**0**	**0**		**0**	**1**	**1**		**0**		

3) Hamming-Bits berechnen (Redundanz erhöhen)

Alle Reihennummern mit dem Bit 1 übernehmen und binär codieren

Reihe 11 =	1	0	1	1
Reihe 6 =	0	1	1	0
Reihe 5 =	0	1	0	1
Anzahl 1-er:	1	2	2	2
Uneven =	1	0	0	0

```
If (Anzahl 1-er = Uneven)
    Then 1
    Else 0
```

4) Hamming-Bits einsetzen (Fehlertoleranz verbessert)

Reihe	12 1100	11 1011	10 1010	9 1001	8 1000	7 0111	6 0110	5 0101	4 0100	3 0011	2 0010	1 0001
Bits	**0**	**1**	**0**	**0**	1	**0**	**1**	**1**	**0**	**0**	**0**	**0**

5) Bits senden (fehlerfrei und mit Fehler)

Bits	**0**	**1**	**0**	**0**	1	**0**	**1**	**1**	**0**	**0**	**0**	**0**

Bits	0	1	1	0	1	0	1	1	0	0	0	0

6) Übermittlung fehlerfrei: Empfangene 12-Bits überprüfen

Reihe	12 1100	11 1011	10 1010	9 1001	8 1000	7 0111	6 0110	5 0101	4 0100	3 0011	2 0010	1 0001
Bits	0	1	0	0	1	0	1	1	0	0	0	0

Alle Reihennummern mit dem Bit 1 binär codieren:

Reihe	11	=	1	0	1	1
Reihe	8	=	1	0	0	0
Reihe	6	=	0	1	1	0
Reihe	5	=	0	1	0	1
Anzahl 1-er:			2	2	2	2
Uneven		=	0	0	0	0

➔ 0_{10}

```
If (Anzahl 1-er = Uneven)
    Then 1
    Else 0
```

➔ Bei 0_{10}: Kein Übermittlungsfehler! Zeichen kann decodiert werden!

Reihe	12 1100	11 1011	10 1010	9 1001	8 1000	7 0111	6 0110	5 0101	4 0100	3 0011	2 0010	1 0001
Bits	0	1	0	0	1	0	1	1	0	0	0	0

7) 8-Datenbits nach ASCII Tabelle decodieren:

ASCII	0	1	0	0	0	1	1	0

➔ F

6a) Übermittlung mit Fehler: Empfangene 12-Bits überprüfen

Fehler heisst in der Digitaltechnik: eine 1 wird zu einer 0 oder umgekehrt! In diesem Beispiel wurde das Bit 10 bei der Übermittlung gekehrt.

Reihe	12 1100	11 1011	10 1010	9 1001	8 1000	7 0111	6 0110	5 0101	4 0100	3 0011	2 0010	1 0001
Bits	**0**	**1**	**1**	**0**	1	**0**	**1**	**1**	**0**	**0**	**0**	**0**

Alle Reihennummern mit dem Bit 1 binär codieren:

Reihe 11 =	1	0	1	1	
Reihe 10 =	1	0	1	0	
Reihe 8 =	1	0	0	0	
Reihe 6 =	0	1	1	0	
Reihe 5 =	0	1	0	1	
Anzahl 1-er:	3	2	3	2	
Uneven =	1	0	1	0	➔ 10_{10}

```
If (Anzahl 1-er = Uneven)
    Then 1
    Else 0
```

➔ Bei ungleich 0_{10}: Ein Fehler ist bei Bit in Reihe 10_{10} passiert! Korrigieren: 1 wird zu einer 0

Reihe	12 1100	11 1011	10 1010	9 1001	8 1000	7 0111	6 0110	5 0101	4 0100	3 0011	2 0010	1 0001
Bits	**0**	**1**	**0**	**0**	1	**0**	**1**	**1**	**0**	**0**	**0**	**0**

7a) 8-Datenbits nach ASCII Tabelle decodieren:

ASCII	**0**	**1**	**0**	**0**	**0**	**1**	**1**	**0**

➔ F

Spiel

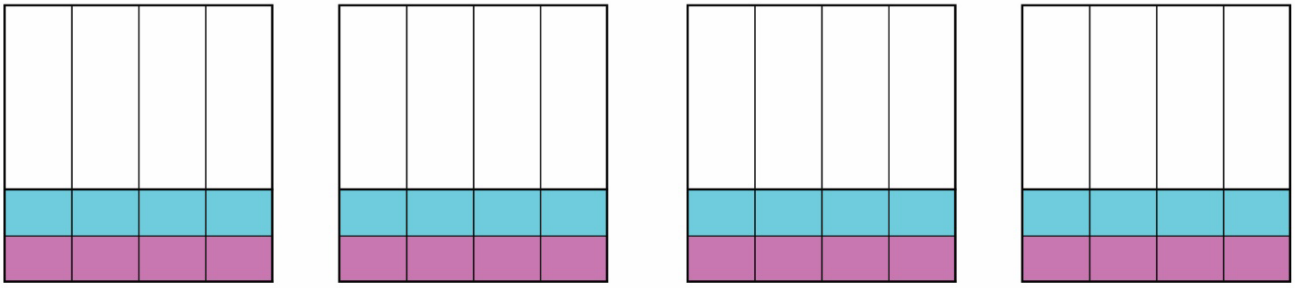
Nun spielen Sie **Sender** und **Empfänger** und dazu bilden Sie 2-er Gruppen!

Anleitung:

1. Schreiben Sie mit Bleistift, so können Sie radieren.
2. **Schreiben** Sie ein Wort mit 4 Buchstaben in die grünen Felder der **Sender-Tabelle** (von oben nach unten)
3. Aus der **ASCII-Tabelle** lesen Sie den Binär-Code jedes gewählten Buchstabens heraus und tragen diese 0 und 1 in die pinken Felder in der jeweiligen Zeile der **Sender-Tabelle (Codieren)** und lassen Sie diese Codierung von einem anderen Spieler (nicht Ihr Empfänger) überprüfen.
4. **Entscheiden** Sie sich für Even oder Uneven und kreisen Sie Ihre Entscheidung auf dem Blatt ein.
5. **Berechnen** Sie für jede Zeile die 4 Hamming-Bits und tragen Sie diese in den blauen Felder der **Sender-Tabelle** ein. Dafür können Sie die 4 leeren weissen Tabellen verwenden.
6. Lassen Sie die fertig ausgefüllte **Sender-Tabelle** vom Lehrer **kontrollieren**.
7. **Übertragen** Sie nun die 12-Bits der pinken und blauen Felder der **Sender-Tabelle** in die pinken Felder der **Empfänger-Tabelle**, wobei Sie in der 2,3 und 4 pro Zeile maximal **einen Fehler** (ein 0 anstatt eine 1 oder umgekehrt) eintragen. Zeile eins übertragen Sie fehlerlos.
8. **Falten** sie das Blatt der gestrichelten Linie entlang, so dass die **Sender-Tabelle nicht mehr sichtbar** ist.
9. **Tauschen** Sie das Blatt mit Ihrem Kollegen / Ihrer Kollegin.
10. **Überprüfen** Sie nun jede Zeile der **Empfänger-Tabelle** und **berechnen** Sie falls nötig die **Korrektur**.
11. **Markieren** Sie in der **Empfänger-Tabelle** die Spalten mit den **Hamming-Bits** mit einem blauen Leuchtstift.
12. **Decodieren** Sie die 8-Datenbits und schreiben Sie den Buchstaben in die grünen Felder der **Empfangs-Tabelle**.
13. **Entfalten** Sie nun das Blatt und **vergleichen** Sie die 4 Buchstaben in der **Sender-Tabelle** mit den 4 Buchstaben der **Empfänger-Tabelle**. Falls diese nicht übereinstimmen suchen Sie zusammen mit dem Kollegen / der Kollegin den Fehler!
14. Beantworten Sie folgende Fragen für sich:
    a. Auf was muss ich beim Codieren / Hamming-Bits berechnen besonders achten?
    b. Worauf muss ich bei der Korrektur-Berechnung und dem Decodieren besonders achten?
    c. Was darf mit nicht mehr passieren?

Sender-Tabelle:

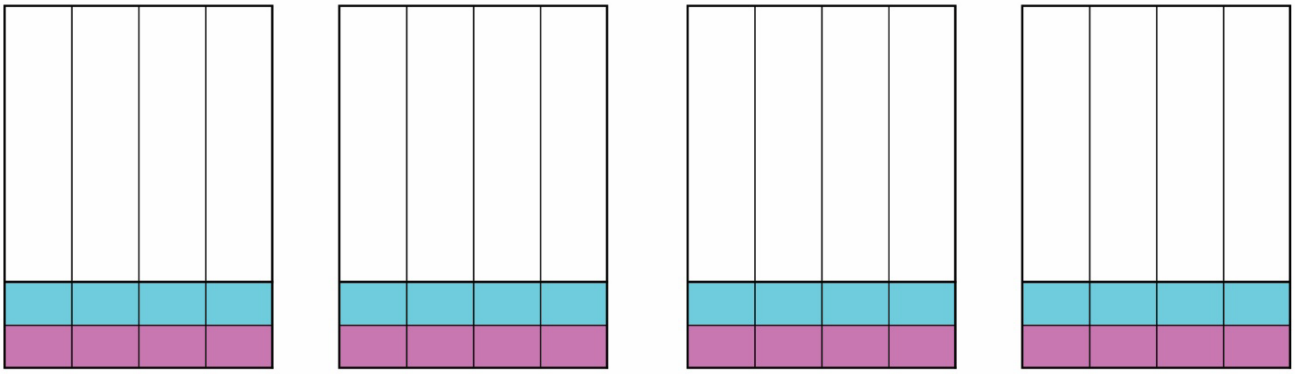
Reihe	12 1100	11 1011	10 1010	9 1001	8 1000	7 0111	6 0110	5 0101	4 0100	3 0011	2 0010	1 0001
ASCII												



Even oder Uneven

Empfänger-Tabelle:

Reihe	12 1100	11 1011	10 1010	9 1001	8 1000	7 0111	6 0110	5 0101	4 0100	3 0011	2 0010	1 0001
ASCII												



Binär				Bit 7		0	0	0	0	0	0	0	0
				Bit 6		0	0	0	0	1	1	1	1
				Bit 5		0	0	1	1	0	0	1	1
				Bit 4		0	1	0	1	0	1	0	1
Bit 3	Bit 2	Bit 1	Bit 0	Hex		0	1	2	3	4	5	6	7
					Dezi	0	16	32	48	64	80	96	112
0	0	0	0	0	0	NUL	DLE		0	@	P	`	p
0	0	0	1	1	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	2	2	STX	DC2	"	2	B	R	b	r
0	0	1	1	3	3	EXT	DC3	#	3	C	S	c	s
0	1	0	0	4	4	EOT	DC4	$	4	D	T	d	t
0	1	0	1	5	5	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	6	6	ACK	SYN	&	6	F	V	f	v
0	1	1	1	7	7	BEL	ETB	'	7	G	W	g	w
1	0	0	0	8	8	BS	CAN	(	8	H	X	h	x
1	0	0	1	9	9	HAT	EM	)	9	I	Y	i	y
1	0	1	0	A	10	IF	SUB	*	:	J	Z	j	z
1	0	1	1	B	11	VT	ESC	+	;	K	[	k	{
1	1	0	0	C	12	FF	FS	,	<	L	\	l	|
1	1	0	1	D	13	CR	GS	-	=	M	]	m	}
1	1	1	0	E	14	SOH	RS	.	>	N	^	n	~
1	1	1	1	F	15	SI	US	/	?	O	_	o	DEL

ASCII	Binär	Hex	Dez
F	0100 ' 0110	46	70

Binär	Octal	Dezimal	Hexadezimal
0000	0	0	0
0001	1	1	1
0010	2	2	2
0011	3	3	3
0100	4	4	4
0101	5	5	5
0110	6	6	6
0111	7	7	7
1000	10	8	8
1001	11	9	9
1010	12	10	A
1011	13	11	B
1100	14	12	C
1101	15	13	D
1110	16	14	E
1111	17	15	F

7	6	5	4	3	2	1	0
128	64	32	16	8	4	2	1